Supplementary material:
Inverse problem regularization with hierarchical variational autoencoders

Code

The code for this project can be found on https://github.com/jprost76/PnP-HVAE

Summary

This supplementary material contains:

- proofs of the theoretical results of the main paper in section 8
- additional implementation details in section 9
- a discussion on the contractivity of the autoencoder and its fixed points in section 10
- additional comparisons with the competing methods in section 11

8. Proofs of the main results

In this section we provide proofs relative to Algorithm 1, Proposition 2, Proposition 3 and the characterization of the fixed point given by Algorithm 2.

8.1. Global minimum of the hierarchical Gaussian negative log-likelihood

In this section we show that under certain conditions Algorithm 1 actually computes the global minimum of $J_2(\boldsymbol{x}, \boldsymbol{z})$ with respct to $\boldsymbol{z}$. To reach that conclusion we first decompose the objective function into several terms (equation (22) in proposition 4). Since many of these terms do not depend on $\boldsymbol{z}$ we conclude that

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}) + B(\boldsymbol{z}).$$

Furthermore, since the second term $(B(\boldsymbol{z}))$ only depends on $\boldsymbol{z}$ via the determinant of the encoder and decoder covariances, we have that under assumption 1

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}).$$

Finally, proposition 5 shows that a functional of the form $A(\boldsymbol{z})$ reaches its global minimum exactly at the point $E_{\boldsymbol{\tau}}(\boldsymbol{x})$ computed by Algorithm 1. Hence under assumption 1 we have that

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x},\boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}) = E_{\boldsymbol{\tau}}(\boldsymbol{x}).$$

Proposition 4 *The objective $J_2(\boldsymbol{x},\boldsymbol{z})$ in equation (17) can be decomposed as*

$$J_2(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x}) - \log p_{data}(\boldsymbol{x}) + A(\boldsymbol{z}) + B(\boldsymbol{z}) + C \tag{22}$$

where

$$A(\boldsymbol{z}) := \sum_{l=0}^{L-1} A_l(\boldsymbol{z}_l, \boldsymbol{z}_{<l}) \tag{23}$$

$$B(\boldsymbol{z}) := \sum_{l=0}^{L-1} B_l(\boldsymbol{z}_{<l}) \tag{24}$$

$$C := \sum_{l=0}^{L-1} C_l \tag{25}$$

and

$$A_l(\boldsymbol{z}_l, \boldsymbol{z}_{<l}) := \|\boldsymbol{z}_l - m_l(\boldsymbol{z}_{<l})\|^2_{S_l^{-1}(\boldsymbol{z}_{<l})} \tag{26}$$

$$B_l(\boldsymbol{z}_{<l}) := \frac{1}{2}\log\det(S_l^{-1}(\boldsymbol{z}_{<l})) + \frac{1}{2}(1-\lambda_l)\log\det(S_{p,l}^{-1}(\boldsymbol{z}_{<l})) \tag{27}$$

$$C_l := \frac{d_l}{2}(\log\lambda_l - \lambda_l\log(2\pi)) \tag{28}$$

and

$$m_{p,l}(\boldsymbol{z}_{<l}) := \mu_{\theta,l}(\boldsymbol{z}_{<l}) \qquad S_{p,l}(\boldsymbol{z}_{<l}) := \Sigma_{\theta,l}^{-1}(\boldsymbol{z}_{<l}) \tag{29}$$

$$m_{q,l}(\boldsymbol{z}_{<l}) := \mu_{\phi,l}(\boldsymbol{z}_{<l}) \qquad S_{q,l}(\boldsymbol{z}_{<l}) := \Sigma_{\phi,l}^{-1}(\boldsymbol{z}_{<l}) \tag{30}$$

$$m_l(\boldsymbol{z}_{<l}) := S_{q,l}(\boldsymbol{z}_{<l})m_{q,l}(\boldsymbol{z}_{<l}) + \lambda_l S_{p,l}(\boldsymbol{z}_{<l})m_{p,l}(\boldsymbol{z}_{<l}) \qquad S_l(\boldsymbol{z}_{<l}) := S_{q,l}(\boldsymbol{z}_{<l}) + \lambda_l S_{p,l}(\boldsymbol{z}_{<l}) \tag{31}$$

Proof. First observe that $q_\phi(\boldsymbol{z}_l|\boldsymbol{x},\boldsymbol{z}_{<l})$ and $p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})$ are multivariate Gaussians as stated in equation (8). Also $p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})^{\lambda_l}$ behaves like a Gaussian with a different normalization constant, namely

$$p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})^{\lambda_l} = \mathcal{N}(\boldsymbol{z}_l; m_{p,l}, \lambda_l^{-1}S_{p,l}^{-1})D_l$$

where the missing normalization constant is

$$D_l = (2\pi)^{-\frac{d_l}{2}(1-\lambda_l)}\lambda_l^{-\frac{d_l}{2}}\det(S_{p,l}^{-1})^{-\frac{1}{2}(1-\lambda_l)}$$

Now $q_\phi(\boldsymbol{z}_l|\boldsymbol{x}, \boldsymbol{z}_{<l})p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})^{\lambda_l}$ is the product of two Gaussians times the correcting term D_l. Since the product of two Gaussians is a Gaussian we obtain

$$q_\phi(\boldsymbol{z}_l|\boldsymbol{x}, \boldsymbol{z}_{<l})p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})^{\lambda_l} = \mathcal{N}(\boldsymbol{z}_l; m_l, S_l^{-1})D_l$$

with mean and variance given by equation (31). Taking $-\log$ in the previous expression, we get $A_l + B_l + C_l$ by grouping into A_l the terms depending on both $\boldsymbol{z}_l$ and $\boldsymbol{z}_{<l}$, in B_l those depending only on $\boldsymbol{z}_{<l}$, and into C_l the constant terms. □

Assumption 1 (Volume-preserving covariances) *The covariance matrices of the HVAE have constant determinant (not depending on $\boldsymbol{z}_{<l}$, although this constant may depend on the hierarchy level l)*

$$\det(\Sigma_{\phi,l}(\boldsymbol{z}_{<l}, \boldsymbol{x})) = c_l(\boldsymbol{x}) \tag{32}$$
$$\det(\Sigma_{\theta,l}(\boldsymbol{z}_{<l})) = d_l \tag{33}$$

Proposition 5 (Algorithm 1 computes the global minimum of $J_2(\boldsymbol{x}, \boldsymbol{z})$ with respect to $\boldsymbol{z}$) *Under Assumption 1 minimizing $J_2(\boldsymbol{x}, \boldsymbol{z})$ w.r.t. $\boldsymbol{z}$ is equivalent to minimizing $A(\boldsymbol{z})$ defined in equations (23) and (26),* i.e.

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}).$$

In addition the global minimum of $A(\boldsymbol{z}_0, \dots, \boldsymbol{z}_{L-1})$ is given by the recursion computed by Algorithm 1, namely:

$$\begin{cases} \boldsymbol{z}_0^\star &= m_0 \\ \boldsymbol{z}_l^\star &= m_l(\boldsymbol{z}_{<l}^\star) \quad \textit{for } l \in \{1, \dots, L-1\} \end{cases} \tag{34}$$

where $\boldsymbol{z}_{<l}^\star = (\boldsymbol{z}_0^\star, \dots, \boldsymbol{z}_{l-1}^\star)$, and $m_l(\boldsymbol{z}_{<l})$ as defined in equations (29) to (31). Put another way, $\boldsymbol{z}^\star = E_{\boldsymbol{\tau}}(\boldsymbol{x})$ as computed by Algorithm 1.

Proof. According to the decomposition of J_2 into several terms (equation (22) in proposition 4), we observe that many of these terms do not depend on $\boldsymbol{z}$. Therefore we conclude that

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}) + B(\boldsymbol{z}).$$

Furthermore, since the second term $(B(\boldsymbol{z}))$ only depends on $\boldsymbol{z}$ via the determinant of the encoder and decoder covariances, and these determinants do not depend on $\boldsymbol{z}$ under assumption 1, we conclude the first part of the proposition, namely

$$\arg\min_{\boldsymbol{z}} J_2(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{z}} A(\boldsymbol{z}).$$

Now let's find the global minimum of $A(\boldsymbol{z})$.

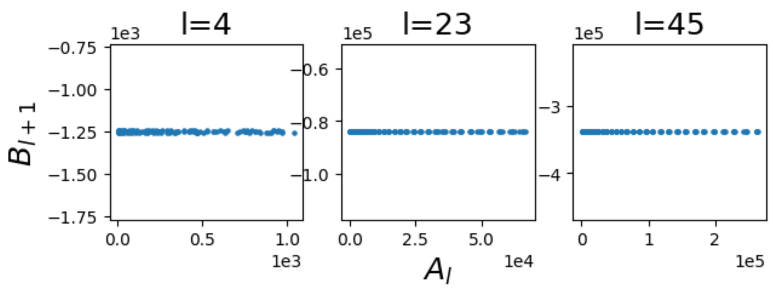


Figure 8: Evolution of $B_{l+1} = \log\det S_{l+1}^{-1}(z_{<l+1})$ as a function of the distance $A_l = \|z_l - \mu_l(z_{<l})||^2_{S_l^{-1}(z_{<l})}$. (experiment made on VDVAE).

It is clear that $A(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{L-1}) \geq 0$ for all $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{L-1}$. It is also simple to verify that:

$$\begin{aligned}&A(\boldsymbol{z}_1^\star, \ldots, \boldsymbol{z}_{L-1}^\star)\\ =&||m_0 - m_0||^2_{S_1^{-1}} + \sum_{l=1}^{L-1} ||m_l(\boldsymbol{z}_{<j}^\star) - m_l(\boldsymbol{z}_{<j}^\star)||^2_{S_l^{-1}(\boldsymbol{z}_{<l}^\star)}\\ =&0.\end{aligned} \tag{35}$$

Therefore the minimum value of A is reached in $\boldsymbol{z}^\star$. Furthermore, for any $\boldsymbol{z} \neq \boldsymbol{z}^\star$, let us denote by k the first value in $\{0, \ldots, L-1\}$ such that $\boldsymbol{z}_k \neq m_k(\boldsymbol{z}_{<k}^\star)$. Then,

$$A(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{L-1}) \geq ||\boldsymbol{z}_k - m_k(\boldsymbol{z}_{<k}^\star)||^2_{S_k^{-1}(\boldsymbol{z}_{<k}^\star)} > 0, \tag{36}$$

which implies that $\boldsymbol{z}^\star$ is the unique minimum of A. □

Discussion on assumption 1 (volume preserving covariance) We showed in proposition 5 that, under assumption 1, Algorithm 1 computes the global minimum of $J_2(\boldsymbol{x}, \boldsymbol{z})$ with respect to $\boldsymbol{z}$. When optimizing z_l in (22) we only consider the impact of $\boldsymbol{z}_l$ on the distance to the Gaussian mean in $A(\boldsymbol{z})$, while ignoring its impact on the covariance volumes in the subsequent levels in the terms $B_{l'}(\boldsymbol{z}_{<l'})$, for $l' > l$. If the covariance volumes are constant as stated in assumption 1, the value of $\boldsymbol{z}_l$ has no impact on the covariance volumes of the subsequent levels, and algorithm 1 gives the global minimizer of $J_2(\boldsymbol{x}, .)$ with respect to $\boldsymbol{z}$. In practice, the HVAE we use does not enforce the covariance matrices of $p(\boldsymbol{z}_l|\boldsymbol{z}_{<l})$ and $q(\boldsymbol{z}_l|\boldsymbol{z}_{<l}, \boldsymbol{x})$ to have constant volume. However, the experiment in figure 8 shows that the variation of $B_{l+1}(\boldsymbol{z}_{<l+1})$ is negligible in front of $A_l(\boldsymbol{z}_l)$. Hence, we can reasonably expect algorithm 1 to yield the minimum of $J_2(\boldsymbol{x}, \boldsymbol{z})$ with respect to $\boldsymbol{z}$. For future works, we could explicitly enforce assumption 1 in the HVAE design.

8.2. Proof of Proposition 2(Lipschitz constant of one iteration)

Proof. For a decoder with constant covariance $\Sigma_\theta^{-1}(\boldsymbol{z}) = \frac{1}{\gamma^2}\,\mathrm{Id}$, we have:

$$\mathrm{T}(\boldsymbol{x}) = \left(A^tA + \frac{\sigma^2}{\gamma^2}\,\mathrm{Id}\right)^{-1}\left(A^t\boldsymbol{y} + \frac{\sigma^2}{\gamma^2}\mu_\theta\left(E_\tau\left(\boldsymbol{x}\right)\right)\right) \tag{37}$$

and then :

$$||T(\mathbf{u}) - T(\mathbf{v})|| \leq \left|\left|\left(A^tA + \frac{\sigma^2}{\gamma^2}\,\mathrm{Id}\right)^{-1}\right|\right| \frac{\sigma^2 L_\tau}{\gamma^2} ||\mathbf{u} - \mathbf{v}||. \tag{38}$$

To conclude the proof, we use that for an invertible matrix M, $||M^{-1}|| = \frac{1}{\sigma_{\mathbf{min}}(\mathbf{M})}$, where $\sigma_{\mathbf{min}}(\mathbf{M})$ is the smallest eigenvalue of M. We also use the fact that α is an eigenvalue of $A^tA + \frac{\sigma^2}{\gamma^2}\,\mathrm{Id}$ if and only if $\alpha = \lambda + \frac{\sigma^2}{\gamma^2}$ for an eigenvalue $\lambda \geq 0$ of the positive definite matrix A^tA. □

8.3. Proof of Proposition 3 (fixed point of PnP-HVAE)

Proof. $\boldsymbol{x}^*$ is a fixed point of T if and only if $\boldsymbol{x}^* = T(\boldsymbol{x}^*)$. Recalling the definition of $\mathrm{T}(\boldsymbol{x}) := \mathrm{prox}_{\gamma^2} f\left(\mathrm{HVAE}\left(\boldsymbol{x}, \boldsymbol{\tau}\right)\right)$, and the definition of proximal operator $\mathrm{prox}_{\gamma^2 f}(\boldsymbol{x}) = \arg\min_{\boldsymbol{t}} \gamma^2 f(\boldsymbol{u}) + \frac{1}{2}||\boldsymbol{x} - \boldsymbol{t}||^2$, the fixed point condition is equivalent to

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{t}} \frac{1}{2}\|\boldsymbol{t} - \mathrm{HVAE}\left(\boldsymbol{x}^*, \boldsymbol{\tau}\right)\|^2 + \gamma^2 f(\boldsymbol{t}).$$

Since f is convex the above condition is equivalent to

$$\boldsymbol{x}^* - \mathrm{HVAE}\left(\boldsymbol{x}^*, \boldsymbol{\tau}\right) + \gamma^2 \nabla f(\boldsymbol{x}^*) = 0.$$

Rearranging the terms we obtain equation (21). □

Under mild assumptions the above result can be restated as follows: $\boldsymbol{x}^*$ is a fixed point of T if and only if

$$\nabla f(\boldsymbol{x}^*) + \nabla g(\boldsymbol{x}^*) = 0,$$

i.e. whenever $\boldsymbol{x}^*$ is a *critical point* of the objective function $f(\boldsymbol{x}) + g(\boldsymbol{x}) = -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$, where the tempered prior is defined as the marginal

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) = \int p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$$

of the joint tempered prior defined in equation (10).

This is shown in the next section.

8.4. Fixed points are critical points

In this section we characterize fixed points of Algorithm 2 as critical points of a posterior density (a necessary condition to be a MAP estimator), under mild conditions. Before we formulate this caracterization we need to review in more detail a few facts about HVAE training, temperature scaling and our optimization model.

HVAE training. In section 3.1 we introduced how VAEs in general (and HVAEs in particular) are trained. As a consequence an HVAE embeds a joint prior

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) := p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z}) \tag{39}$$

from which we can define a marginal prior on $\boldsymbol{x}$

$$p_\theta(\boldsymbol{x}) := \int p_\theta(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}. \tag{40}$$

In addition, from the ELBO maximization condition in (5) and Bayes theorem we can obtain an alternative expression for the joint prior, namely

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}). \tag{41}$$

Temperature scaling. After training we reduce the temperature by a factor $\boldsymbol{\tau}$, which amounts to replacing $p_\theta(\boldsymbol{z})$ by

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}) := \prod_{l=0}^{L-1} \frac{p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{<l})^{\frac{1}{\tau_l^2}}}{\tau_l^{\frac{d_l}{2}}}$$

as shown in equation (10), leading to the joint tempered prior

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}, \boldsymbol{z}) := p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}). \tag{42}$$

The corresponding marginal tempered prior on $\boldsymbol{x}$ becomes

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) := \int p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \tag{43}$$

and the corresponding posterior is

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) := p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}, \boldsymbol{z}) / p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}). \tag{44}$$

The joint tempered prior also has an alternative expression (based on the encoder). Indeed substituting $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ from equations (39) and (41) into (42) we obtain

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}, \boldsymbol{z}) = \frac{p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z})}{p_\theta(\boldsymbol{z})} q_\phi(\boldsymbol{z}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}). \tag{45}$$

Substituting this result into definition (44) we obtain an alternative expression for the tempered posterior

$$p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) / p_\theta(\boldsymbol{z}). \tag{46}$$

Optimization model. Since we are using a scaled prior $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ encoded in our HVAE to regularize the inverse problem, the ideal optimization objective we would like to minimize is

$$U(\boldsymbol{x}) := \underbrace{-\log p(\boldsymbol{y}|\boldsymbol{x})}_{f(\boldsymbol{x})} \underbrace{-\log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})}_{g(\boldsymbol{x})}. \tag{47}$$

Since $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ is intractable our algorithm seeks to minimize a relaxed objective (see equation (15)). Nevertheless, under certain conditions (to be specified below) this is equivalent to minimizing the ideal objective (47).

Fixed-point characterization. We start by characterizing $\nabla \log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ in terms of an HVAE-related denoiser (Proposition 6). Then we relate this denoiser to the quantity $\mathrm{HVAE}(\boldsymbol{x}, \tau)$ that is computed by our algorithm (Proposition 7). As a consequence we obtain that the fixed point condition in Proposition 3 can be written as $\nabla U(\boldsymbol{x}) = 0$ (see Corollary 2).

Proposition 6 (Tweedie's formula for HVAEs.) *For an HVAE with Gaussian decoder* $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \mu_\theta(\boldsymbol{z}), \gamma^2 I)$*, the following denoiser based on the HVAE with tempered prior*

$$D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) := \int \mu_\theta(\boldsymbol{z})\, p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \tag{48}$$

satisfies Tweeedie's formula

$$D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) - \boldsymbol{x} = \gamma^2 \nabla \log p_{\theta,\boldsymbol{\tau}}(x) = -\gamma^2 \nabla g(\boldsymbol{x}). \tag{49}$$

Proof. From the definition of $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ in equation (43) we have that

$$\nabla \log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) = \frac{1}{p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})} \int \nabla_{\boldsymbol{x}} p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}) d\boldsymbol{z}.$$

From the pdf of the Gaussian decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ its gradient writes

$$\nabla_{\boldsymbol{x}} p_\theta(\boldsymbol{x}|\boldsymbol{z}) = -\frac{1}{\gamma^2}(\boldsymbol{x} - \mu_\theta(\boldsymbol{z})) p_\theta(\boldsymbol{x}|\boldsymbol{z}).$$

Replacing this in the previous equation we get

$$\begin{aligned}
\nabla \log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) &= \frac{1}{\gamma^2} \int (\mu_\theta(\boldsymbol{z}) - \boldsymbol{x}) \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z})}{p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})} d\boldsymbol{z} \\
&= \frac{1}{\gamma^2} \int (\mu_\theta(\boldsymbol{z}) - \boldsymbol{x}) p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\
&= \frac{1}{\gamma^2} \left(\int \mu_\theta(\boldsymbol{z})\, p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} - \boldsymbol{x} \right).
\end{aligned}$$

In the second step we used the definitions of the joint tempered prior $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x},\boldsymbol{z})$ (42) and the tempered posterior $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x})$ (44). The last step follows from the fact that $\int p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z} = 1$ according to definitions (44) and (43). Finally applying the definition of the denoiser $D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ in the last expression we obtain Tweedie's formula (49). □

Under suitable assumptions the denoiser defined above coincides with $\mathrm{HVAE}(\boldsymbol{x},\boldsymbol{\tau})$ computed by our algorithm.

Assumption 2 (Deterministic encoder) *The covariance matrices of the encoder defined in equation (8) are 0,* i.e. $\Sigma_{\phi,l}(\boldsymbol{z}_{<l},\boldsymbol{x}) = 0$ *for* $l = 0,\ldots,L-1$. *Put another way* $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \delta_{E_{\boldsymbol{\tau}}(\boldsymbol{x})}(\boldsymbol{z})$ *is a Dirac centered at* $E_{\boldsymbol{\tau}}(\boldsymbol{x})$*.*

Proposition 7 *Under Assumption 2 the function* $\mathit{HVAE}(\boldsymbol{x},\boldsymbol{\tau})$ *computed by Algorithm 2 coincides with the denoiser* $D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ *defined in equation (48).*

Proof. $\mathrm{HVAE}(\boldsymbol{x},\boldsymbol{\tau})$ is defined in Proposition 3 as

$$\mathrm{HVAE}(\boldsymbol{x},\boldsymbol{\tau}) = \mu_\theta\left(E_\tau(\boldsymbol{x})\right).$$

First observe that for a deterministic encoder we also have $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) = \delta_{E_{\boldsymbol{\tau}}(\boldsymbol{x})}(\boldsymbol{z})$. Indeed for any test function h:

$$\begin{aligned}\int h(\boldsymbol{z})p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z} &= \int h(\boldsymbol{z})q_\phi(\boldsymbol{z}|\boldsymbol{x})p_{\text{data}}(\boldsymbol{x})/p_\theta(\boldsymbol{z})d\boldsymbol{z}\\ &= h(E_{\boldsymbol{\tau}}(\boldsymbol{x}))\underbrace{p_{\text{data}}(\boldsymbol{x})/p_\theta(E_{\boldsymbol{\tau}}(\boldsymbol{x}))}_{Z(\boldsymbol{x})}.\end{aligned}$$

And the normalization constant $Z(\boldsymbol{x})$ should be equal to 1 because $\int p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z} = Z(\boldsymbol{x}) = 1$. Hence $p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \delta_{E_{\boldsymbol{\tau}}(\boldsymbol{x})}(\boldsymbol{z})$.

Finally applying the definition of $D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$ we obtain

$$\begin{aligned}D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) &= \int \mu_\theta\left(\boldsymbol{z}\right)p_{\theta,\boldsymbol{\tau}}(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{z} = \mu_\theta\left(E_{\boldsymbol{\tau}}(\boldsymbol{x})\right)\\ &= \mathrm{HVAE}(\boldsymbol{x},\tau).\end{aligned}$$

□

Combining Propositions 7, 3 and 6 we obtain a new characterization of fixed points as critical points.

Corollary 2 *Under Assumption 2* $\boldsymbol{x}^*$ *is a fixed point of* T *if and only if*

$$\nabla f(\boldsymbol{x}^*) + \nabla g(\boldsymbol{x}^*) = 0 \tag{50}$$

where $g(\boldsymbol{x}) = -\log p_{\theta,\boldsymbol{\tau}}(\boldsymbol{x})$*.*

Proof. From Proposition 6 we have that

$$-\nabla g(\boldsymbol{x}) = \frac{1}{\gamma^2}\left(D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) - \boldsymbol{x}\right).$$

From Proposition 7 we have that (under Assumption 2) $D_{\theta,\boldsymbol{\tau}}(\boldsymbol{x}) = \mathrm{HVAE}(\boldsymbol{x},\tau)$. In combination with the previous result:

$$-\nabla g(\boldsymbol{x}) = \frac{1}{\gamma^2}\left(\mathrm{HVAE}(\boldsymbol{x},\tau) - \boldsymbol{x}\right).$$

Finally, Proposition 3 allows to conclude that

$$-\nabla g(\boldsymbol{x}) = \nabla f(\boldsymbol{x}).$$

□

9. Details on PatchVDVAE architecture

In this section, we provide additional details about the architecture of PatchVDVAE. Then, we present the choice of the hyperparameters used for the concurrent methods (presented in section 6 of the main paper) .

9.1. PatchVDVAE

Figure 9 provides a detailed overview of the structure of a PatchVDVAE network. The architecture follows VDVAE model [1], except for the first top-down block, in which we replace the constant input by a latent variable sampled from a Gaussian distribution. The architecture presented in figure 9 illustrates the structure of HVAE networks, but the number of blocks is different to the PatchVDVAE network used in our experiments. Our PatchVDVAE top-down path is composed of $L = 30$ top-down blocks of increasing resolution. The image features are upsampled using an unpooling layer every 5 blocks. The first unpooling layer performs a $\times 4$ upsampling, and the following unpooling layers perform $\times 2$ upsampling. The dimension of the filters is 256 in all blocks. In order to save computations in the residual blocks, the 3×3 convolutions are applied on features of reduced channel dimension (divided by 4). 1×1 convolutions are applied before and after the 3×3 convolutions to respectively reduce and increase the number of channels. The latent variables $\boldsymbol{z}_l$ are tensors of shape $12 \times H_l \times W_l$, where the resolution H_l, W_l corresponds to the resolution of the corresponding top-down-block. The bottom-up network structure is symmetric to the top-down network, with 5 residual blocks for each scale, and pooling layers between each scale.

9.2. Hyperparameters of compared methods

Face image restoration. For ILO, we found that optimizing the first 5 layers of the generative network offered the best trade-off between image quality and consistency with the observation.

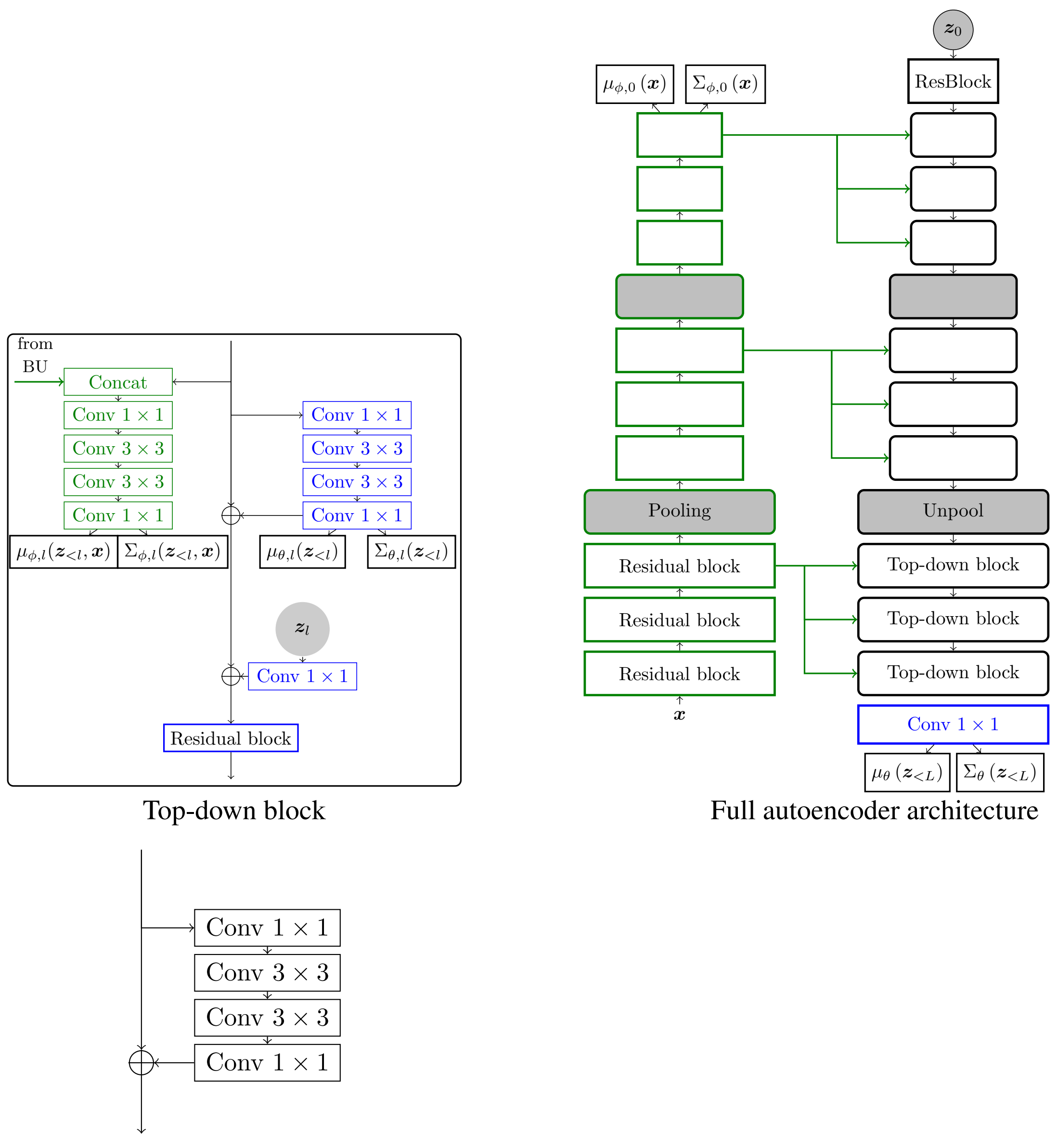


Figure 9: Structure of the PatchVDVAE architecture. For clarity, we omit the non-linearity after each convolution.

Hence, we optimize the 5 first layers for 100 iterations each. This choice is different from the official implementation, where they only optimize the 4 first layers for a lower number of iterations, trading restoration performance for speed. For DPS, we set the scale hyper-parameter ζ' (described in subsection C.2 in[2]) to $\zeta' = 1$ for the deblurring and super-resolution experiments reported in this paper.

Natural images restoration - Deblurring. For the three tested methods, we use the official implementation provided by the authors, along with the pretrained models. For EPLL, we use the default parameters in the official implementation.

For GS-PnP, using the notation of the paper, we use the suggested hyperparameter $\lambda_\nu = 0.1$ for the motion blur kernels and $\lambda_\nu = 0.75$ for the Gaussian kernels.

For PnP-MMO, we use the denoiser trained on $\sigma_{den} = 0.007$. On deblurring with $\sigma = 2.55$ we use the default parameters in the implementation. for higher noise levels ($\sigma = 7.65$; $\sigma = 12.75$), and we set the strength of the gradient step as $\gamma = \sigma_{den}/(2\sigma||h||)$, where h corresponds to the blur kernel.

Natural images restoration- Inpainting. For EPLL, we use the default parameters provided in the authors matlab code. For GS-PnP, after a grid-search, we chose to set $\lambda_\nu = 1$ and $\sigma_{denoiser} = 10$.

10. Discussion on the conctractivity of HVAE

We showed in section 5 that PnP-HVAE converges to a fixed point under the assumption that $\boldsymbol{x} \to \text{HVAE}(\boldsymbol{x}, \tau)$ is contractive. If this condition is met, the sequence of u_k defined by $u_{k+1} = HVAE(u_k, \tau)$ should converge to a fixed point. Figure 10 presents the **evolution of a fixed point iteration** $u_{k+1} = HVAE(u_k, \tau)$. The image is smoothened over the iterations, and finally converges to a piececewise constant image. We used patchVDVAE for this experiment.

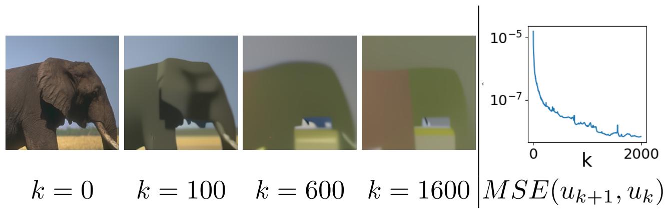


Figure 10: Fixed-point iterations of patchVDVAE for $\tau = 0.99$.

11. Comparisons

In this section, we provide additional visual results on face images and natural images.

11.1. Additional results on face image restoration

We provide additional comparisons with the GAN-based ILO method on inpainting (figure 11), $\times 4$ super-resolution (figure 12) and deblurring (figure 13). PnP-HVAE provides equally or more plausible glasses in the first column) inpaiting than ILO. For superresolution, ILO produces sharper but not realistics faces. This is an agreement with the scores presented in table 1). For deblurring, ILO creates textures on faces that looks realistic (low LPIPS) but are less consistent with the observation (significantly lower PSNR and SSIM).

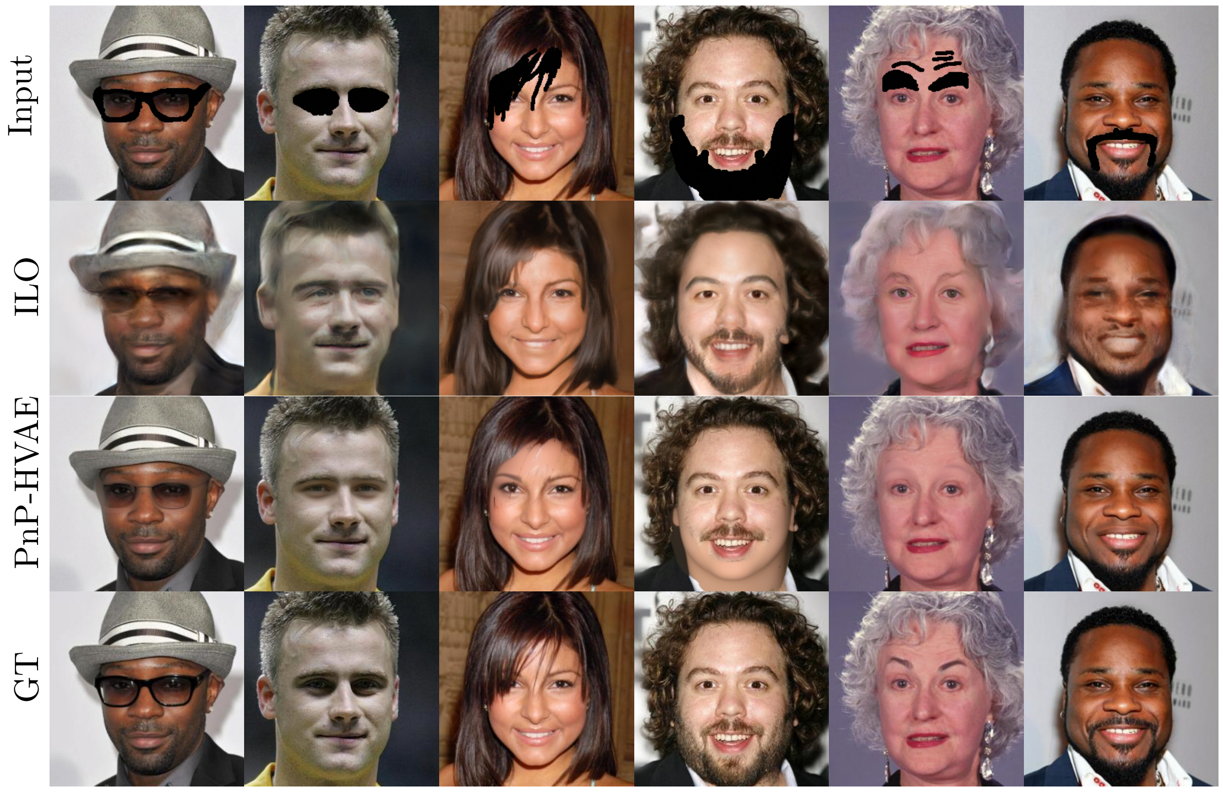


Figure 11: Inpainting

11.2. Additional results on natural images restoration

We finally present additional results on natural images restoration. All the PnP-HVAE images presented below were produced using our PatchVDVAE model. We also provide visual comparisons with concurrent PnP methods and EPLL. For deblurring (figures 15 and 16, PnP methods perform better than EPLL. Following quantitative results of figure 2, for larger noise level, PnP-HVAE outperforms PnP-MMO and provides restoration close to GS-PnP.

For inpainting (figure 17), the hierarchical structure of PatchVDVAE leads to more plausible reconstructions, and PnP-HVAE outperforms the compared methods.

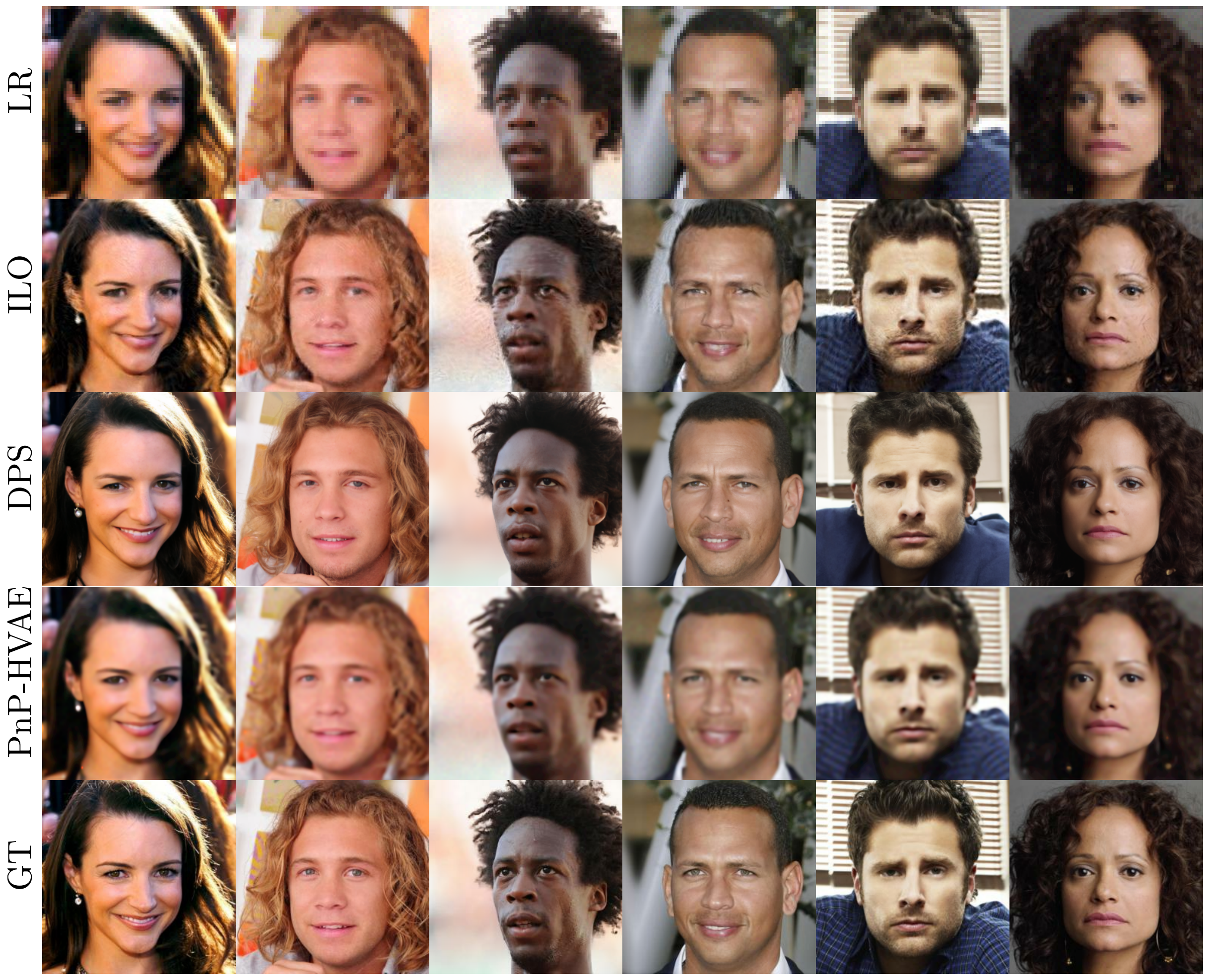


Figure 12: $\times 4$ super-resolution, with kernel (a) from Figure 14 and $\sigma = 3$

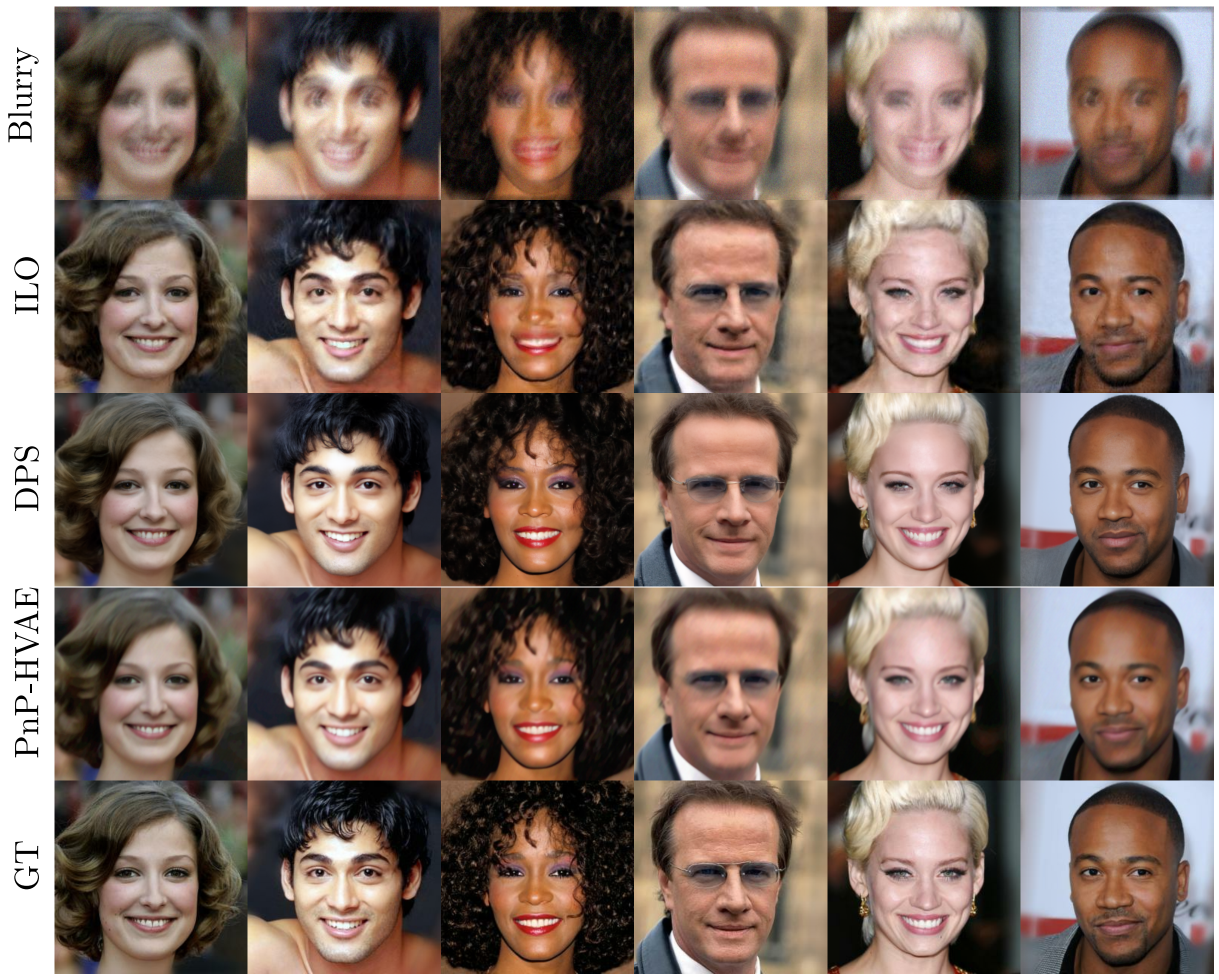


Figure 13: Deblurring, with kernel (d) from Figure 14 and $\sigma = 8$

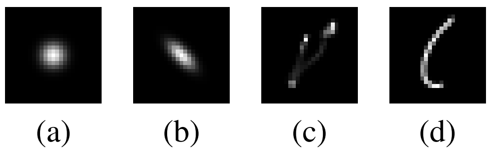
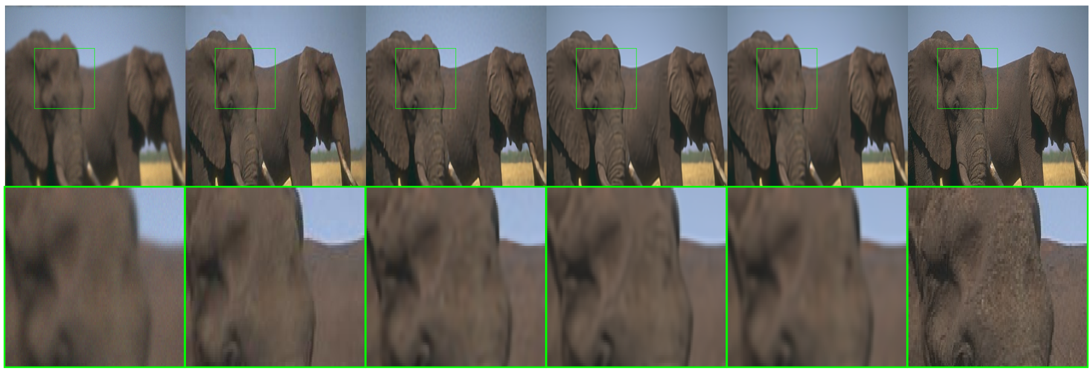
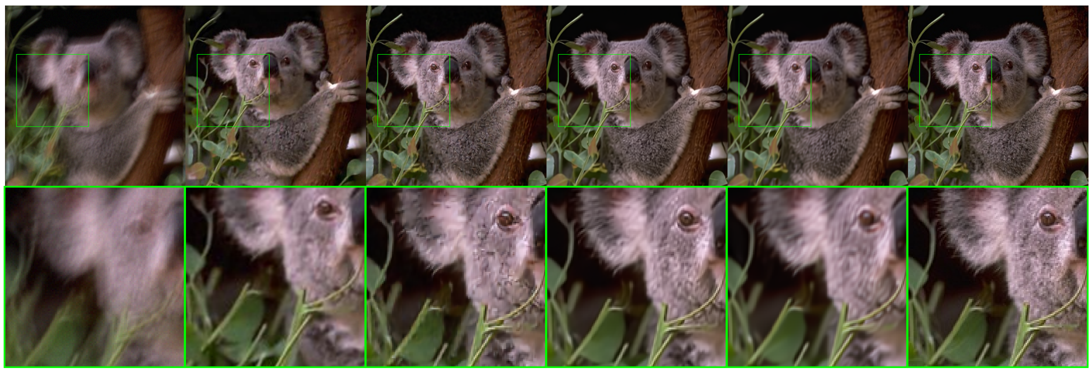


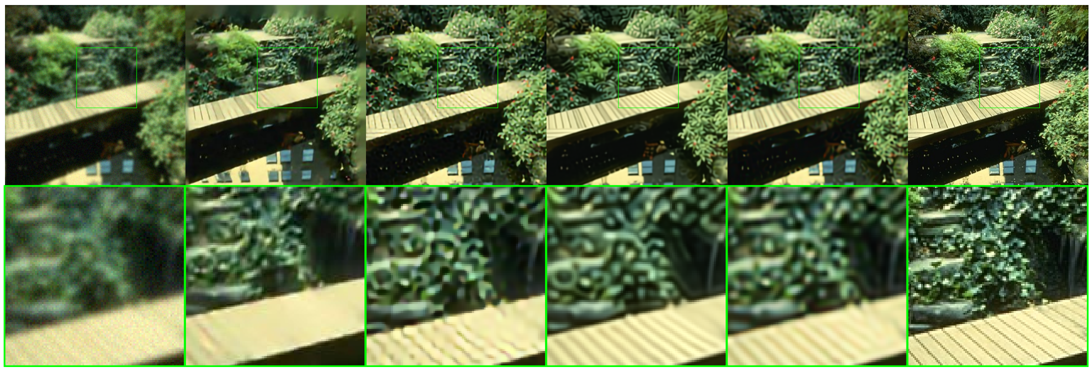
Figure 14: Kernels used for deblurring experiments, from [3]



(a) kernel (a), $\sigma = 2.55$

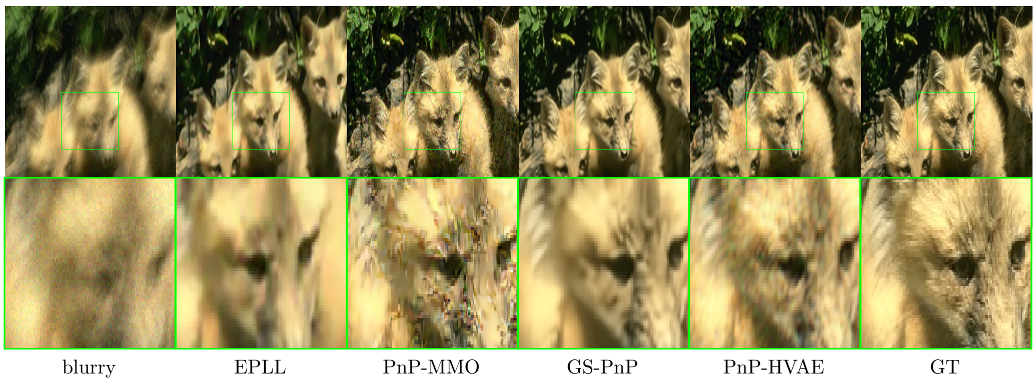


(b) kernel (c), $\sigma = 2.55$

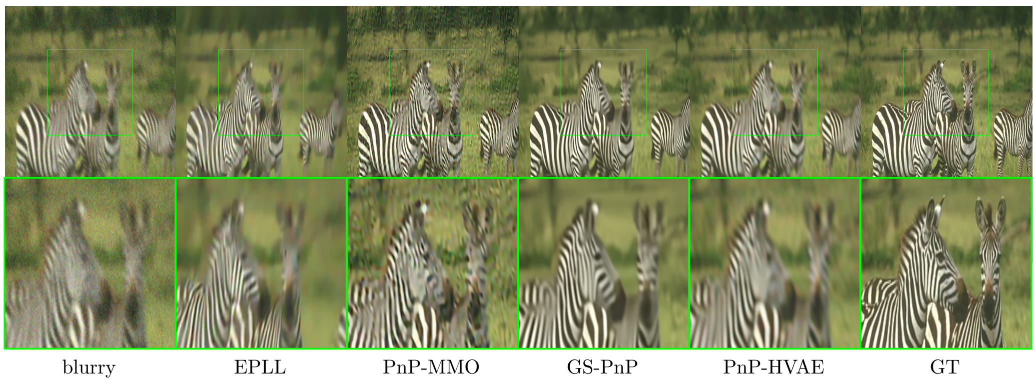


(c) kernel (a), $\sigma = 7.65$

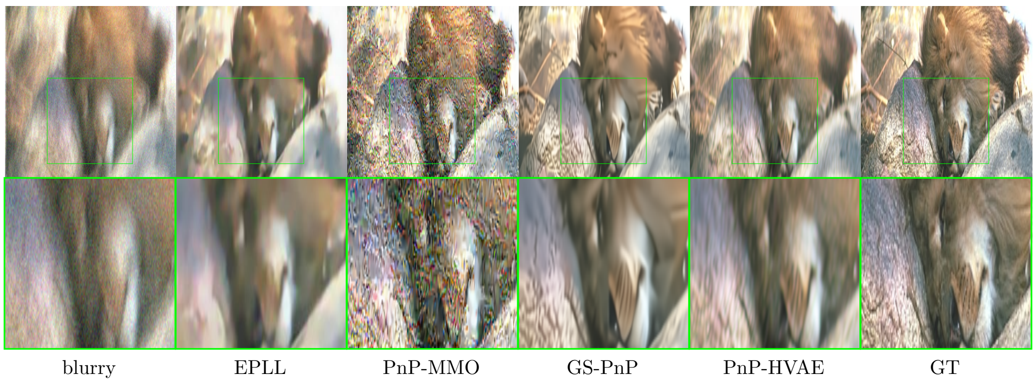
Figure 15: Deblurring results on BSD



(a) kernel (d), $\sigma = 7.65$



(b) kernel (b), $\sigma = 12.75$



(c) kernel (d), $\sigma = 12.75$

Figure 16: Deblurring results on BSD

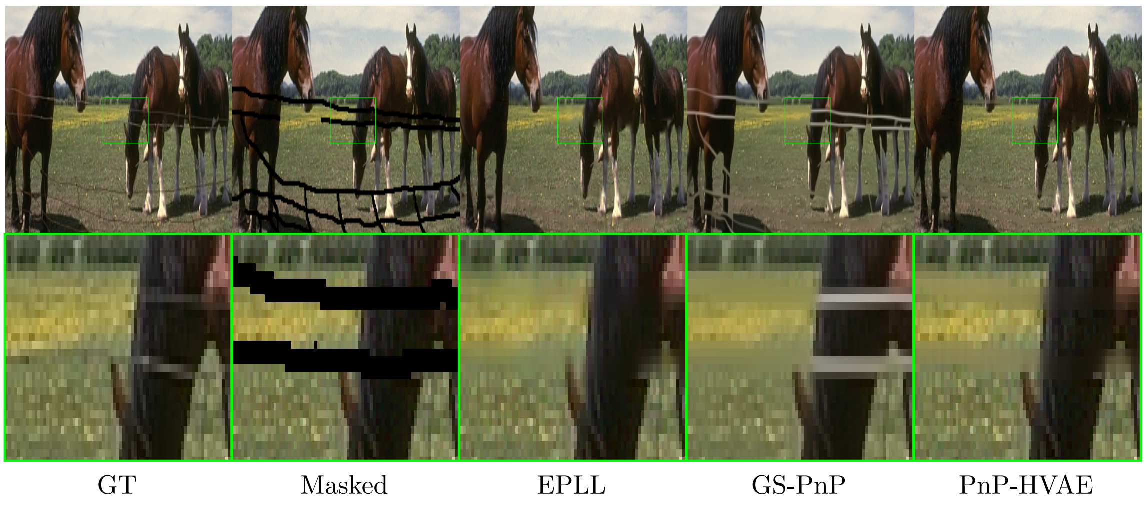


Figure 17: Natural images inpainting

References

[1] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2020. 9

[2] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2022. 10

[3] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1964–1971. IEEE, 2009. 14